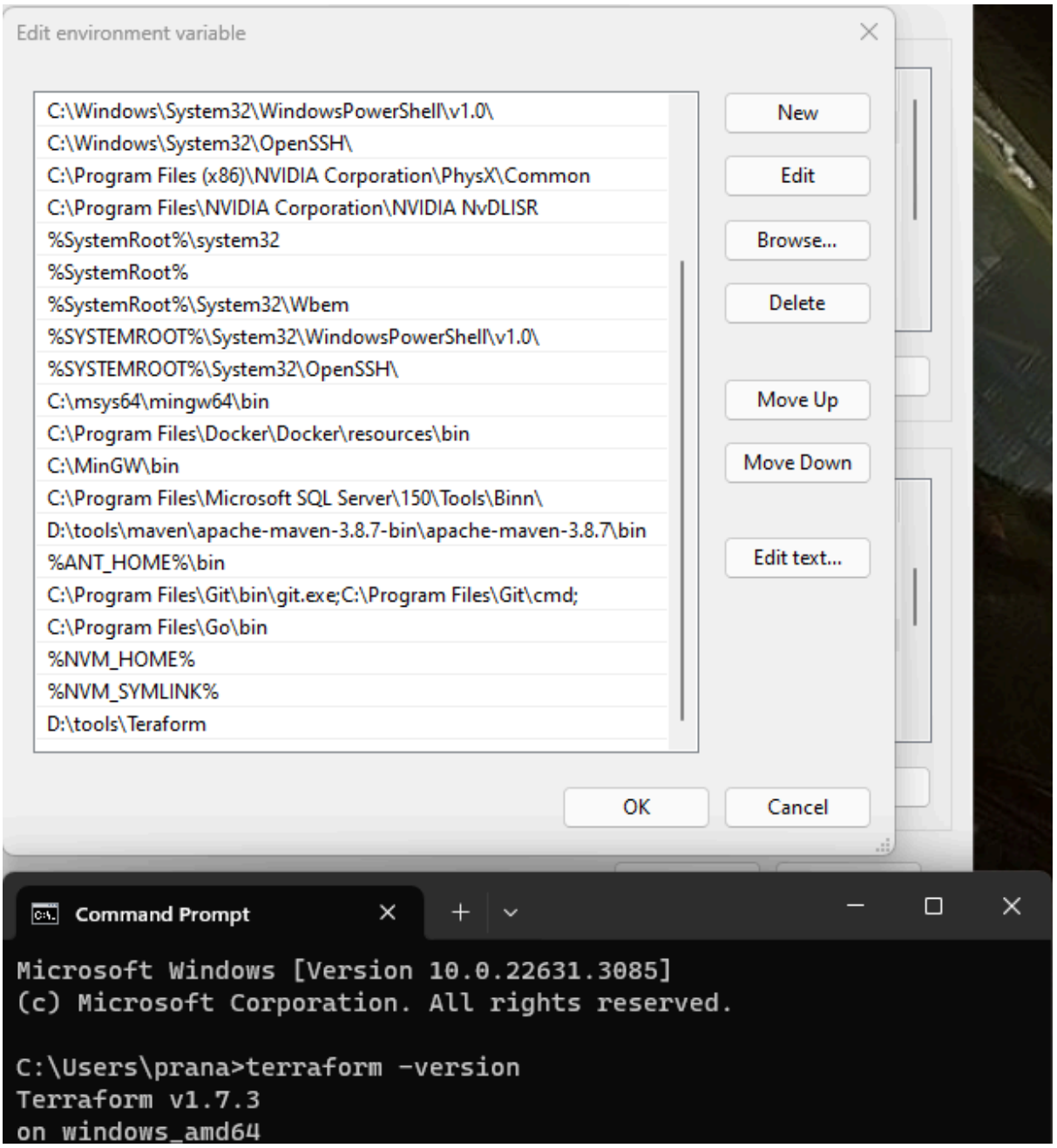
SUBMITTED BY: Pranay Mayal, B2 (NON HONS.)

Lab Exercise 1– Install Terraform on Windows

Step 1: Download Terraform File for Windows.

Step 2: Add Terraform Path to System Environment Variables.



Step 3: Verify Windows Terraform Installation.

SUBMITTED BY: Pranay Mayal, B2 (NON HONS.)

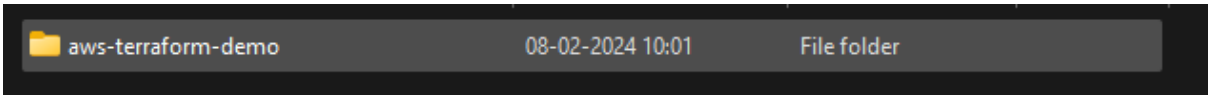
Lab Exercise 2– Terraform AWS Provider and IAM User Setting

Prerequisites:

- Terraform Installed: Make sure you have Terraform installed on your machine. Follow the official installation guide if needed.
- AWS Credentials: Ensure you have AWS credentials (Access Key ID and Secret Access Key) configured. You can set them up using the AWS CLI or by setting environment variables.

Procedure:

Step 1: Create a New Directory



Step 2: Create Terraform Configuration File (main.tf):

```
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.31.0"
}
}
}

provider "aws" {
region = "ap-south-1"
```

Step 3: Initialize Terraform:

```
PS D:\Sem 6 DevOps\SPCM\Lab\My Lab Files and PDFS\Lab Exercise 2\aws-terraform-demo> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```